Introduction to optimization with applications in astronomy and astrophysics

Stephane Canu, Rémi Flamary, David Mary

Introduction to optimization with applications in astronomy and astrophysics

Stéphane Canu, Rémi Flamary, David Mary

July 18, 2016

Abstract

This chapter aims at providing an introduction to numerical optimization with some applications in astronomy and astrophysics. We provide important preliminary definitions that will guide the reader towards different optimization procedures. We discuss three families of optimization problems and describe numerical algorithms allowing, when this is possible, to solve these problems. For each family, we present in detail simple examples and more involved advanced examples. As a final illustration, we focus on two worked-out examples of optimization applied to astronomical data. The first application is a supervised classification of RR-Lyrae stars. The second one is the denoising of galactic spectra formulated by means of sparsity inducing models in a redundant dictionary.

1 Optimization in astronomy and astrophysics

Optimization can be sketched as the art of finding a 'best compromise'. This is a two-step art. The first step translates the problem, the desired objective and possibly some constraints into a mathematical form. The second step finds the best compromise that can be achieved within the framework defined by the first step.

In Astrophysics, the information of interest can rarely be observed directly. A first reason for this is that raw data like time series or astronomical images (galaxies, nebulae, cosmological microwave background,...) are only exploitable once corrected for various distortions[1] (foreground nuisances sources, atmospheric perturbations, limited resolution of the telescope, instrumental noise sources, etc.). A second reason is that the astrophysical information to be detected or estimated is often not directly related to the data. For instance, information like the mass of an extrasolar planet, the distance to its host star, or even simply its very existence can sometimes be revealed only by means of sophisticated detection and estimation algorithms in time series [Perryman, 2011] or images [Starck et al., 2015]. Information like the internal structure of pulsating stars, their age, radius and effective temperature can be evaluated by

[1] See for instance the lectures by C. Ferrari and J.-F. Cardoso at BasMatI school.

comparing the detected oscillation modes with numerical models [Christensen-Dalsgaard, 2003, 2008]. A third reason is that modern astronomy leads to build instruments giving access to surveys of increasingly larger scale, like the Gaia satellite[2], the Large Synoptic Survey Telescope[3] or the Square Kilometer Array[4]. The number of astrophysical sources captured by these surveys, in the billions, prevents from performing a dedicated analysis of each individual source. Instead, the extraction of information needs to be statistical and automated with, for instance, detection and classification pipelines.

Consequently, the astrophysical information always results from a complex, more or less supervised extraction process. Because formalizing the extraction process by means of objectives and constraints is an efficient way to proceed, the path of optimization is useful for various studies tackled from the perspective of inverse problems, data mining, or machine learning. In practice, optimization is at the crossroads of several methodological families and plays therefore a central role in many astronomical studies.

In the rest of this section we sketch some simple examples of optimization problems that can occur in astronomy and astrophysics. We will express these problems under the general form:

$$\min_{\mathbf{x}\in\mathcal{C}} \quad F(\mathbf{x}) \tag{1}$$

In (1), F is the objective function. The best compromise is formulated as a minimization problem (as for instance for a data fit problem). The vector $\mathbf{x} = [x_1, \ldots, x_n]^t$ collects the n optimization (or primal) variables and $\mathcal{C} \subset \mathbb{R}^n$ is a convex set. The variables can be image parameters (pixel intensity values for instance) or coefficients representing the data in some transform domain (wavelet coefficients for instance). The notion of convexity is very important in optimization. First, because existence and unicity of the solution to a convex problem are usually guaranteed. Second, convex problems can be solved efficiently with generic algorithms, of which various implementations are freely available online. Sec. 2.1. provides definition and intuitive representation of convexity.

1.1 Image reconstruction

The best example of information restitution in Astronomy is perhaps the need for counteracting the 'telescope blur'. Indeed, the (Fourier) frequential contents of an image observed through a telescope is modified. When the telescope's effect is the same whatever the position of the source (i.e., when the observing system is translation invariant), the image is the result of a convolution of the true sky intensity distribution by a band pass (usually lowpass for monolithic telescopes) filter of impulse response called Point Spread Function (PSF). Because the width of the PSF is inversely proportional to the diameter of the telescope, bigger

[2] `http://sci.esa.int/gaia/28820-summary/`
[3] `http://lsst.org`
[4] `https://www.skatelescope.org`

telescopes provide images with finer details. To reconstruct images with a better resolution, a deconvolution step is often applied to data images. The aim of this step is to restore some of the high frequencies switched-off by the telescope, to obtain (hopefully) an image closer to the original image than the data image.

When the geometry of the telescope is known exactly, the PSF can be computed theoretically and the convolution can be exactly modeled by a linear operator on the vectorized image, $\mathbf{x}$. The optimization problem becomes

$$\min_{\mathbf{x}\geq\mathbf{0}} \quad L(\mathbf{y}, \mathbf{H}\mathbf{x}) \tag{2}$$

where optimization acts here directly on the flux values in each pixel, $\mathbf{H}$ is the linear modeled obtained from the PSF (convolution matrix), L measures the discrepancy between the observation $\mathbf{y}$ and the convolved image $\mathbf{H}\mathbf{x}$. The notation $\mathbf{x} \geq \mathbf{0}$ means that each variable x_i is contrained to be nonnegative as it corresponds to an intensity.

This problem has been studied in depth in the inverse problem community. In practice, the data fidelity function L can be selected according to a statistical model of the measurement noise, or simply to make the computations more tractable. For instance, when the noise on $\mathbf{H}\mathbf{x}$ is additive and Gaussian, minimizing $L(\mathbf{y}, \mathbf{H}\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$ is equivalent to maximum likelihood estimation. When the noise can be modeled as Poisson noise (corresponding to photon counts), then the maximum likelihood estimator is obtained by minimizing $L(\mathbf{y}, \mathbf{H}\mathbf{x}) = \sum_i y_i \log\left(\frac{y_i}{(\mathbf{H}\mathbf{x})_i}\right) + (\mathbf{H}\mathbf{x})_i - y_i$, that is, the Kullback-Leibler divergence between $\mathbf{y}$ and $\mathbf{H}\mathbf{x}$. [Lantéri et al., 2013].

The optimization problem (2) with positivity constraints can be solved with the Image Space Reconstruction Algorithm (ISRA) [De Pierro, 1987] for the euclidean data fitting and by the well known Richardson-Lucy [Richardson, 1972, Lucy, 1974] Algorithm when using KL divergence[5].

Optimization techniques dedicated for image reconstruction in radio-astronomy are currently subject to very active developments with the advent of various precursors and pathfinders of the future SKA. Radio telescopes aim at having a better resolution using antennas separated by large distances. In radioastronomy, the image is actually formed *a posteriori* by computing numerically Fourier Transform of data recorded on each antenna (as opposed to direct imaging formed in the focal plane of classical optical telescopes by means of mirrors and/or lenses). In the Fourier domain, the bandpass of a radiotelescope array has however a very sparse support, owing to the limited number of antennas [Thompson et al., 2008]. New generation radiotelescopes, for which (few and costly) antennas are replaced by (numerous and cheap) dipoles, can improve this situation. Besides *resolution*, the two other advantages of new radiotelescopes arrays are their flux *sensitivity* (e.g., SKA will use millions of dipoles, allowing a large collected flux) and their capability of being pointed electronically and not physically (as dipoles virtually see the whole sky, introducing electronically and off-line a delay and summing coherently the signals of two dipoles make

[5]Note that care must be taken with negative data values in these algorithms.

them sensitive to one direction of the sky). This is why they are called 'software telescopes'. This capability, called *survey speed*, allows to cover large parts of the sky much more rapidly than classical dish-based radio arrays.

In this domain a classical approach is the CLEAN algorithm proposed by Högbom [1974] and its variants [Cornwell, 2008, Wakker and Schwarz, 1988] that can be applied to images with diffuse intensity distributions. Such approaches belong to the family of sparse *greedy* algorithms. In these algorithms, sparsity is imposed by selecting iteratively some model(s) among a collection of possible models. The selected models are those who decrease most the data fidelity term. Hence, the approach of greedy algorithms is different from global optimisation strategies discussed in this course, which focus on the best compromise (lowest cost) with respect to a cost function including data fidelity and sparsity promoting terms (as in (4) for instance).
As far as (global) optimization is concerned, the last decade has witnessed many new radio imaging algorithms based on convex optimization with additional regularization terms such as sparsity in overcomplete dictionaries [Carrillo et al., 2012, 2014]. In such cases, the problem (2) is augmented with additional terms, which are usually non-differentiable to promote sparsity. Non-differentiability requires to use dedicated algorithms (based for instance on proximal operators) and some of them will be discussed below. As a final note, it is important to emphasize that the approaches discussed here can be extended to spatio-spectral reconstruction, i.e., reconstruction of cubes of images collected in different colors. This is a particularly active research field for SKA for instance, which will provide large images over hundreds of channels.

1.2 Spectral object detection and denoising

Astronomical observations often measure not only a quantity of light but also a full spectrum of the object of interest. This leads to large datasets of spectra associated to galaxies or stars, such as the Sloan Digital Sky Survey [York et al., 2000]. Novel observation techniques such as the Multi Unit Spectroscopic Explorer (MUSE) provide 3D hyperspectral images at high spectral and spatial resolutions. The high spectral resolution comes with an important noise level and spectral denoising is critical for some sources. This problem can also be expressed as an optimization problem.

For instance in Bourguignon et al. [2012, 2010] the spectrum is modeled as a sum of a sparse vector (impulses) and a continuous vector that is sparse in the Discrete Cosine Transform (DCT) basis leading to the following optimization problem, given $\lambda_l > 0$ and $\lambda_c > 0$

$$\min_{\mathbf{x}\in\mathbb{R}^n} \ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda_l \|\mathbf{x}^l\|_1 + \lambda_c \|\mathbf{F}\mathbf{x}^c\|_1 \quad \text{s.t. } \mathbf{x} = \mathbf{x}^l + \mathbf{x}^c, \tag{3}$$

where $\mathbf{F}$ is the linear operator that computes the DCT of the spectrum. Efficient algorithms such as proximal methods can be used to solve (3).

1.3 Machine learning in astronomy

Ball and Brunner [2010] propose a review of machine learning applications to astronomy, defined as the science of *building computer programs that automatically improve with data [Mitchell, 2006].* A good reference is also the book Ivezić et al. [2014] that illustrates several possible applications. Machine learning techniques have been used to automatically detect gravitational lenses in images by Agnello et al. [2015] and variable stars in Gaia data by Süveges et al. [2015].

The improvement process of machine learning is often formalized as an optimization problem. For instance when learning an automatic classifier that will for instance detect gravitational lenses in an image [Agnello et al., 2015], one minimize the following optimization problem, for a given $\lambda > 0$

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad L(\mathbf{x}) + \lambda\Omega(\mathbf{x}), \tag{4}$$

where L is a data fitting term that measures the error of a given model $\mathbf{x}$ on the available training data set and Ω is a regularization term that promotes a simple classifier with good generalization abilities that is the ability to perform good prediction on unseen data. A typical choice would be the least square error for L and some norm for Ω.

Among the most known classifiers that fit this optimization problem, one can cite Support Vector Machines, that have shown state of the art performances in several domains, penalized logistic regression or neural networks whose last layer of neurons is estimated by solving problem (4). Theory and examples on such classifiers will be provided below.

The remaining of this chapter is organised as follows: The overall framework of optimization is introduced in the next section with a definition of the notions of convexity and differentiability. Then, using these distinctions in the three following sections, different kinds of optimization problems of increasing complexity are presented and illustrated by examples. They cover the cases of differentiable unconstrained optimization (section 3), differentiable constrained optimization (section 4) and non-differentiable unconstrained optimization (section 5). Section 6 presents two applications involving astronomical data and machine learning.

2 Optimization framework

All four optimization problems presented in the previous section (1), (2), (3) and (4) can be put in the following general setting

$$\mathcal{P} = \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & F(\mathbf{x}) & \\ \text{with} & h_j(\mathbf{x}) = 0 & \forall j = 1,\dots,p \\ \text{and} & g_i(\mathbf{x}) \leq 0 & \forall i = 1,\dots,q. \end{cases} \tag{5}$$

Function F is referred as the objective function while functions h_j and g_i define respectively the equality and inequality constraints. The *domain* or *feasible set* of problem $\mathcal{P}$ is the set of vectors $\mathbf{x}$ fulfilling the constraints, that is

$$\big\{\mathbf{x} \in \mathbb{R}^n \;\big|\; h_j(x) = 0,\ \forall j = 1, \ldots, p \text{ and } g_i(x) \leq 0,\ \forall i = 1, \ldots, q\big\}.$$

In problem (2) $p = 0$, $q = n$ and function $g_i(\mathbf{x}) = -x_i$. In problem (3) $p = 1$ and $q = 0$ with function $h_1(\mathbf{x}, \mathbf{x}^l, \mathbf{x}^c) = \mathbf{x} - \mathbf{x}^l - \mathbf{x}^c$. Problem (4) is said to be unconstrained since $p = q = 0$. In problem (1) the constraint is formulated as the inclusion in a convex set, an important notion at which we now give a closer look.

2.1 Convexity

A convex set can be defined as a set $\mathcal{C}$ such that, for any two points $\mathbf{x}, \mathbf{y} \in \mathcal{C}^2$, $\mathbf{z} \in \mathcal{C}$ for every point $\mathbf{z} = \alpha\mathbf{x} + (1-\alpha)\mathbf{y}$, with $0 \leq \alpha \leq 1$. An important example of convex set are polyhedra defined as the solution set of a finite number of linear inequalities of the form $\mathbf{Hx} \leq \mathbf{b}$ for a given matrix $\mathbf{H}$ and vector $\mathbf{b}$. Constraints of problems (2) and (3), (i.e., respectively $\mathbf{x} \geq 0$ and $\mathbf{x} = \mathbf{x}^l + \mathbf{x}^c$), are defining convex sets. Together with convex sets, it is useful to define convex functions.

Definition 1. *A function F is said to be convex if it lies below its chords, that is*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \quad F(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha F(\mathbf{x}) + (1-\alpha)F(\mathbf{y}), \text{ with } 0 \leq \alpha \leq 1. \quad (6)$$

A function is said to be strictly convex when the two inequalities above are strict.

Strict convexity implies that the function has a unique minimum.

An illustration of convex and non-convex sets and functions is shown in Figure 1. There exists strong relationships between convex functions and convex sets. For instance, if a function f is convex, then the set $\{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \leq 0\}$ is convex. Hence, a convex set can be represented by a family of convex functions associated with inequalities.

An optimization problem is said to be convex when its objective F and its feasible set are convex. This happens in particular when its inequality constraint functions g_i are convex while its equality constraint functions h_j are affine. In that sense, problems presented in the previous section are convex when F is convex in (1), L is convex in (2), the two norms are convex in (3), and L and Ω are convex in (4). In practice proving or verifying that a problem is convex can be complicated but there exists several rules that can be used. For instance, a positive weighted sum of convex function is convex and a composition of a positive convex function by a non-decreasing function is convex. For a comprehensive list of these rules we refer the reader to [Boyd and Vandenberghe, 2004, Chapter 3].

The importance of convexity in optimization is related to the nature of the underlying issues summarized hereafter. Let $\mathbf{x}^\star$ denote the optimal solution of problem $\mathcal{P}$, that is the point having the smallest value of F among all vectors

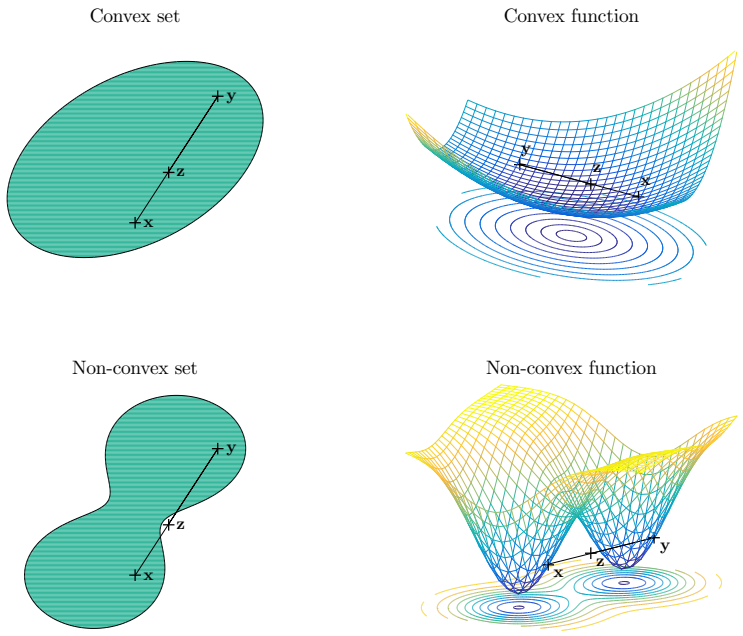

Figure 1: Illustration of convexity on sets (left) and functions (right). The convex (up) and non-convex (down) sets are obtained from the levelset of the 2D functions.

satisfying the constraints. The problem of finding $\mathbf{x}^\star$ rises different issues such as:

- existence and unicity of the solution $\mathbf{x}^\star$,
- necessary and sufficient optimality conditions (how to characterize $\mathbf{x}^\star$),
- computation of $\mathbf{x}^\star$ (the algorithmic issues),
- analysis and reformulation of the problem.

When the optimization problem is convex, existence and unicity of the solution are generally guaranteed. Furthermore, there exist reliable and efficient algorithms for solving convex optimization problems.

Finally, all convex optimization problems are not equivalent: there is a hierarchy of complexity among them. The simplest classes are the classes of linear (LP) and quadratic (QP) programs defined as:

$$\text{(LP)} \quad \left\{ \begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \mathbf{c}^t\mathbf{x} \\ \text{with} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{array} \right. \qquad \text{(QP)} \quad \left\{ \begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \frac{1}{2}\mathbf{x}^t\mathbf{G}\mathbf{x} + \mathbf{c}^t\mathbf{x} \\ \text{with} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{array} \right.$$

A quadratic program (QP) is convex when matrix $\mathbf{G}$ is positive definite. A LP is a particular case of a QP with matrix $\mathbf{G} = \mathbf{0}$. These problems are pretty generic and there exists mature technology to solve them. For a more general framework in convex optimization, one can design its optimization problem in order to ensure convexity using a set of rules called disciplined convex programming (see

Grant et al. [2006]). When the problem is expressed with these rules it can be solved using a generic optimization toolbox such as CVX [Grant and Boyd, 2014].

2.2 Differentiability

Together with convexity, differentiability is another important notion allowing to distinguish among the different types of optimization problems. Indeed, necessary and sufficient optimality conditions are related with Fermat's rule involving the gradient when F is differentiable or else the more general notion of sub-differential.

Assume F is differentiable in the sense that all its partial derivatives $\frac{\partial F}{\partial x_i}$ exist. In that case, its gradient can be defined as follows:

Definition 2 (Gradient)**.** *The gradient $\boldsymbol{\nabla} F(\mathbf{x})$ of a function F at point $\mathbf{x}$ is the vector whose components are the partial derivatives of F.*

Example 1. *(Least square) Given a $p \times n$ design matrix $\mathbf{H}$ and a response vector $\mathbf{y}$, the gradient of the least square cost function $F_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2$ is*

$$\boldsymbol{\nabla} F_1(\mathbf{x}) = \mathbf{H}^t(\mathbf{H}\mathbf{x} - \mathbf{y}).$$

Theorem 1. *If F is convex and differentiable, then*

$$F(\mathbf{x} + \mathbf{h}) \geq F(\mathbf{x}) + \boldsymbol{\nabla} F(\mathbf{x})^t \mathbf{h}, \qquad \forall \mathbf{h} \in \mathbb{R}^n.$$

This means that the gradient can be used to define a linear lower bound of the objective function at point $\mathbf{x}$ and a descent direction. As a consequence, the gradient can be used to characterize the global solution of the unconstrained convex minimization problem with the so-called first order optimality conditions :

$$\mathbf{x}^\star = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \quad F(\mathbf{x}) \qquad \Leftrightarrow \qquad \boldsymbol{\nabla} F(\mathbf{x}^\star) = 0.$$

The gradient provides an optimality condition and a descent direction with handy computation rules: sum and chain rule.

When F is non differentiable the gradient no longer exists. In this case one should use sub-differential instead. Its definition uses the notion of sub-gradient.

Definition 3 (subgradient)**.** *A vector $\mathbf{d} \in \mathbb{R}^n$ is a subgradient of a function F at point $\mathbf{x}$ if*

$$F(\mathbf{x} + \mathbf{h}) \geq F(\mathbf{x}) + \mathbf{d}^t \mathbf{h}, \qquad \forall \mathbf{h} \in \mathbb{R}^n.$$

In other words a sub-gradient is a vector that defines an hyperplane that stay below the function for all $\mathbf{h}$ (see Figure 2).

Definition 4 (subdifferential)**.** *The subdifferential $\partial F(\mathbf{x})$ of a function F at point $\mathbf{x}$ is the set (possibly empty) of all its sub gradients*

$$\partial F(\mathbf{x}) = \left\{ \mathbf{g} \in \mathbb{R}^n \mid F(\mathbf{x} + \mathbf{d}) \geq F(\mathbf{x}) + \mathbf{g}^t \mathbf{d}, \quad \forall \mathbf{d} \in \mathbb{R}^n \right\}.$$

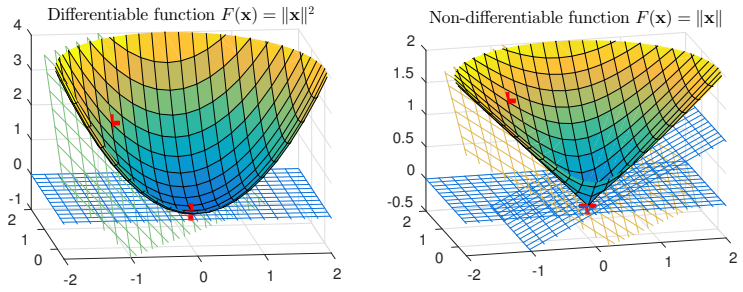

Figure 2: Illustration of gradients and subgradient 2D linear approximation for a differentiable (left) and non-differentiable (right) function at 2 points denoted in red. Note that when the function is differentiable, there exist a unique tangent hyperplane whereas when it is not differentiable such as point $[0, 0]$ at right there exists an ensemble of hyperplanes that stay below the function, two of which are reported in blue on the right plot.

Example 2. *Assuming $n = 1$ we have:*

$F_2(x) = |x|$ $\qquad\qquad$ $\partial F_2(0) = \{g \in \mathbb{R} \mid -1 \leq g \leq 1\}$,

$F_3(x) = \max(0, 1 - x)$ $\qquad\qquad$ $\partial F_3(1) = \{g \in \mathbb{R} \mid -1 \leq g \leq 0\}$.

Convexity of F implies that it has at least one supporting hyperplane at every point of $\mathbb{R}^n$, that is $\partial F(\mathbf{x}) \neq \emptyset$. Furthermore, If F is differentiable, $\boldsymbol{\nabla} F(\mathbf{x})$ is the unique subgradient of F at $\mathbf{x}$ that is

$$\partial F(\mathbf{x}) = \big\{\boldsymbol{\nabla} F(\mathbf{x})\big\}.$$

Finally, the subdifferential characterizes the global solution of a convex problem since in that case (Fermat's Theorem)

$$\mathbf{x}^\star = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} F(\mathbf{x}) \quad \Leftrightarrow \quad \mathbf{0} \in \partial F(\mathbf{x}^\star). \tag{7}$$

2.3 Different types of optimization problems

When F is nondifferentiable and non convex, the sub-differential non longer exists. Its non convex generalization is known as the Clarke subdifferential [Clarke, 1990]. When F is differentiable and non convex, the Fermat's rule becomes only necessary and characterizes a local minimum. The analysis of an optimization problem and the choice of an algorithm to solve it depend on its convexity, its differentiability and whether it is constrained or not.

In the following, we discuss more in detail three different optimization problems with potential applications in astronomy. The unconstrained convex optimization case, the constrained convex optimization where Fermat's rule generalises to Karush-Kuhn-Tucker (KKT) conditions and non differentiable optimization involving convex and non convex specific situations.

Function	Convex	Diff.
$\omega(x) = x^2$ $\Omega(\mathbf{x}) = \|\mathbf{x}\|^2$	✓	✓
$\omega(x) = 1 - \exp^{-x^2}$	✗	✓
$\omega(x) = \|x\|$ $\Omega(\mathbf{x}) = \|\mathbf{x}\|_1$	✓	✗
$\omega(x) = \sqrt{\|x\|}$ $\Omega(\mathbf{x}) = \|\mathbf{x}\|_{1/2}$	✗	✗

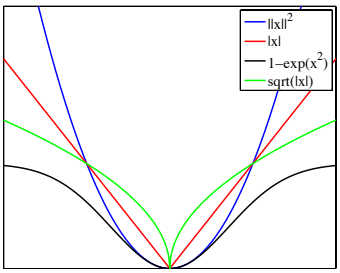

Figure 3: Examples of different type of penalty functions involved in machine learning optimization problems ilustrated on the right [for more examples see Antoniadis et al., 2011, and references therein].

Example 3. *Examples of these different situations can be found considering different regularization functions Ω involved in problem (4). Very often, this penalty term can be expressed as the sum of single variable functions as follows*

$$\Omega(\mathbf{x}) = \sum_{i=1}^{n} \omega(x_i). \tag{8}$$

Figure 3 presents some examples of such regularization terms and their form.

Note that, even in the non convex and non differentiable cases, these functions show some regularity as illustrated figure 3.

3 Unconstrained convex and differentiable optimization

3.1 The theory of unconstrained convex optimization

Consider the following unconstrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}), \tag{9}$$

with F convex. We have seen in the previous section that the Fermat' s rule provides a simple characterization of the minimizers of a function as the zeros of its subdifferential or of its gradient when F is differentiable.

Definition 5 (Gradient descent)**.** *A general setting to solve such a minimization problem amounts at considering the following sequence*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}, \tag{10}$$

where $\mathbf{d}^{(k)} \in \mathbb{R}^n$ is a descent direction such that $\boldsymbol{\nabla} F(\mathbf{x}^{(k)})^t \mathbf{d}^{(k)} < 0$ and $\rho^{(k)} \in \mathbb{R}^+$ the associated stepsize. A natural choice for the descent direction is, when it

exists, the opposite of the gradient $\mathbf{d}^{(k)} = -\boldsymbol{\nabla} F(\mathbf{x}^{(k)})$. *For a good choice of* $\rho^{(k)}$ *and* $\mathbf{d}^{(k)}$ *this sequence converges towards* $\mathbf{x}^\star$ *the solution of the problem. The conditions for convergence are discussed more in details in Bertsekas [1999], Nocedal and Wright [2006]*

The choice of the stepsize $\rho^{(k)}$ at each iteration can be seen as a "line search" since it boils down to finding a real value along the fixed descent direction $\mathbf{d}^{(k)}$ (a line) that provides a sufficient descent. A typical line search method, known as Armijo line search [Nocedal and Wright, 2006, Algorithm 3.1], initializes ρ with a given step and decrease this step by multiplying it until a descent condition is met. Convergence is proven for line search methods that ensures sufficient decrease of the cost function [Nocedal and Wright, 2006, Chapter 3].

Interestingly one can see the gradient descent method as the iterative solving of a local approximation of a function. To illustrate that, we define the following.

Definition 6. *A function* F *is gradient Lispchitz if there exists a constant* L_F *such that*

$$\|\boldsymbol{\nabla} F_1(\mathbf{x}+\mathbf{d}) - \boldsymbol{\nabla} F_1(\mathbf{x})\| \leq L_F \|\mathbf{d}\|, \qquad \forall \mathbf{d} \in \mathbb{R}^n, \forall \mathbf{x} \in \mathbb{R}^n. \tag{11}$$

The constant L_F *is called the Lipschitz constant of* $\boldsymbol{\nabla} F$.

Example 4. *The least square cost function* $F_1(\mathbf{x})$ *is gradient Lispchitz with constant* $L_F = \|\mathbf{H}^t\mathbf{H}\|$. *Indeed* $\boldsymbol{\nabla} F_1(\mathbf{x}+\mathbf{d}) = \mathbf{H}^t(\mathbf{H}\mathbf{x} - \mathbf{y}) + \mathbf{H}^t\mathbf{H}\mathbf{d}$, *so that*

$$\boldsymbol{\nabla} F_1(\mathbf{x}+\mathbf{d}) - \boldsymbol{\nabla} F_1(\mathbf{x}) = \mathbf{H}^t\mathbf{H}\mathbf{d},$$

and

$$\|\boldsymbol{\nabla} F_1(\mathbf{x}+\mathbf{d}) - \boldsymbol{\nabla} F_1(\mathbf{x})\| \leq \|\mathbf{H}^t\mathbf{H}\|\|\mathbf{d}\| = L_F\|\mathbf{d}\|.$$

Note that if F is gradient Lispchitz, we have the following second order majorization of function F around $\mathbf{x}$, also called descent Lemma:

$$F(\mathbf{x}+\mathbf{d}) \leq F(\mathbf{x}) + \boldsymbol{\nabla} F(\mathbf{x})^t\mathbf{d} + \frac{L_F}{2}\|\mathbf{d}\|^2, \qquad \forall \mathbf{d} \in \mathbb{R}^n, \forall \mathbf{x} \in \mathbb{R}^n. \tag{12}$$

For a proof and more details, see [Bertsekas, 1999, Prop. A.24].

If one wants to find a direction $\mathbf{d}^{(k)}$ that minimizes the quadratic approximation above, one computes its gradient *w.r.t.* $\mathbf{d}^{(k)}$ with gives $\boldsymbol{\nabla} F(\mathbf{x}^{(k)}) + L_F\mathbf{d}^{(k)}$. It can be set to $\mathbf{0}$ by choosing $\mathbf{d}^{(k)} = -\frac{1}{L_F}\boldsymbol{\nabla} F(\mathbf{x}^{(k)})$. This procedure allows us to find the gradient descent direction but also gives us a maximum value for the step size $\rho^{(k)} \leq \frac{1}{L_F}$ that ensures an objective value decrease at each iteration, hence convergence.

Now let's assume that F is two times differentiable, in this case it is possible to define its Hessian.

Definition 7 (Hessian)**.** *The Hessian* $\boldsymbol{\nabla}^2 F(\mathbf{x})$ *of a function* F *at point* $\mathbf{x}$ *is the* $n \times n$ *matrix valued function whose components are* $\nabla^2_{ij} F(\mathbf{x}) = \frac{\partial^2 F}{\partial x_i \partial x_j}(\mathbf{x})$ *the second order partial derivatives of* F.

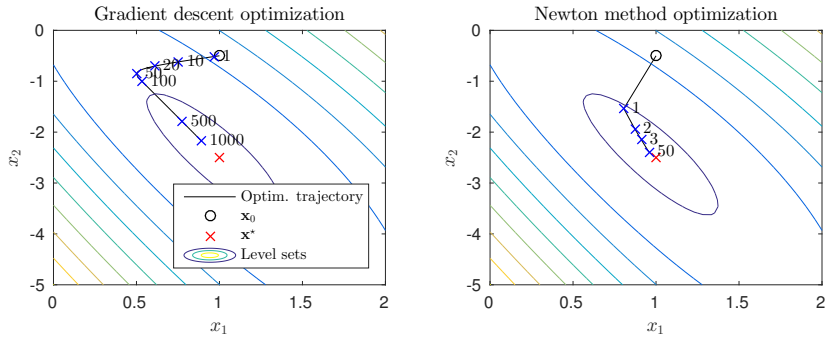

Figure 4: Illustration of gradient descent and Newton methods on a simple 2D logistic regression loss. We also show the iteration number along some samples to illustrate the difference in convergence speed.

Example 5. *The Hessian of F_1 the least square cost function of example 1 is*

$$\boldsymbol{\nabla}^2 F_1(x) = \mathbf{H}^t\mathbf{H}.$$

Note that F is convex if and only if $\forall \mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\nabla}^2 F(\mathbf{x})$ is a positive definite matrix. Using the gradient and the Hessian (when they exist) it is possible to use the local second order Taylor expansion of function F defined by

$$F(\mathbf{x}+\mathbf{d}) = F(\mathbf{x}) + \mathbf{d}^t\boldsymbol{\nabla} F(\mathbf{x}) + \tfrac{1}{2}\mathbf{d}^t\boldsymbol{\nabla}^2 F(\mathbf{x})\mathbf{d} + o(\|\mathbf{d}\|^2). \tag{13}$$

Again this approximation of the function can be used to find a descent direction for a (second order) descent algorithm called the Newton method.

Definition 8 (Newton method)**.** *The Newton method consist in minimizing at each iteration the quadratic Taylor expansion (13) around $\mathbf{x}^{(k)}$. The optimal direction is $\mathbf{d}^{(k)} = -(\boldsymbol{\nabla}^2 F)^{-1}(\mathbf{x}^{(k)})\boldsymbol{\nabla} F(\mathbf{x}^{(k)})$. The resulting algorithm is called the Newton method. It fits the general formula* (10) *with $\rho^{(k)}$ equal one.*

Example 6. *Gradient and Newton iterations for the least square problem* $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho^{(k)}\mathbf{H}^t(\mathbf{H}\mathbf{x}^{(k)} - \mathbf{y})$, *Gradient*

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{H}^t\mathbf{H})^{-1}\mathbf{H}^t(\mathbf{H}\mathbf{x}^{(k)} - \mathbf{y})$. *Newton*

In this particular case, when starting with $\mathbf{x} = 0$, the Newton method converges in one iteration.

The second order Taylor approximation is a far better approximation than the Lispschitz approximation (11) which implies a better convergence speed for Newton methods. Nevertheless it requires the inversion of a $n \times n$ matrix at each iteration which can be untractable for large n. A comparison of gradient and Newton descent on a simple 2D minimization problem is illustrated Figure 4.

Note that there exists a family of optimization methods called quasi-Newton that jointly benefits from the better approximation provided by Newton method

and from an efficient update in the gradient descent. The principle of these methods is to estimate sequentially the Hessian matrix or its inverse using efficient rank-1 updates. One of those approaches avoids the storage of an $n \times n$ matrix and leads to the the popular implementation named Limited-Memory BFGS (after Broyden, Fletcher, Goldfarb and Shanno, see Liu and Nocedal [1989] for details). This approach is considered among the most efficient way to solve differentiable optimization problems and is available in most optimization softwares.

Finally, when dealing with large scale optimization problem, the computation of the full gradient may be untractable. In that case, a stochastic gradient descent should be considered [for more details see for instance Bottou, 2004]. The practical choice of a method depends on a trade off between the computational cost of a single iteration and the convergence speed of the method [see for instance Bubeck, 2015, and included references].

3.2 Advanced example: logistic regression

The binary logistic regression is a popular two class classification method and a nice example of unconstrained optimization [for more details see for instance Hastie et al., 2005]. Given a $p \times n$ design matrix $\mathbf{H}$ and a p dimensional vector of labels $\mathbf{y} \in \{0,1\}^p$, logistic regression amounts to solve the following unconstrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; F_\ell(\mathbf{x}) = \sum_{i=1}^{p} \left(-y_i(\mathbf{H}\mathbf{x})_i + \log\big(1+\exp(\mathbf{H}\mathbf{x})_i\big)\right), \tag{14}$$

with $(\mathbf{H}\mathbf{x})_i$ the i^{th} component for vector $\mathbf{H}\mathbf{x}$. This logistic cost function can be interpreted as the negative log likelihood of a Bernoulli sample $\{y_i, i = 1, p\}$ with parameter $\mathbf{p}_i(\mathbf{x}) = \exp((\mathbf{H}\mathbf{x})_i)/(1+\exp((\mathbf{H}\mathbf{x})_i))$. Problem (14) aim at finding the parameter $\mathbf{x}$ for the probability above that maximize the likelihood on the training data $(\mathbf{H}, \mathbf{y})$. Once parameter $\mathbf{x}$ is estimated the prediction of the class of a new observation $\mathbf{h}$ is done with a simple likelihood ratio test:

$$\Lambda(\tilde{\mathbf{H}}_k) = \tilde{\mathbf{H}}_k\mathbf{x} \quad \begin{matrix} 1 \\ \gtrless \\ 0 \end{matrix} \quad \text{threshold}, \tag{15}$$

where the default value of the threshold is 0.

Since this objective function is the composition of two times differentiable functions, its gradient and Hessian matrix both exist and are

$$\begin{aligned} \boldsymbol{\nabla} F_\ell(x) &= \mathbf{H}^t(\mathbf{p}-\mathbf{y}) \\ \boldsymbol{\nabla}^2 F_\ell(x) &= \mathbf{H}^t\mathbf{W}\mathbf{H}, \end{aligned} \tag{16}$$

with $p_i = \exp(\mathbf{H}\mathbf{x})_i/(1+\exp(\mathbf{H}\mathbf{x})_i)$ and W a diagonal matrix of general term $W_{ii} = p_i(1-p_i), i = 1, n$.

Given the gradient and the Hessian matrix, Newton iterations build the following sequence

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{H}^t\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^t(\mathbf{p}-\mathbf{y}).$$

Algorithm 1 The Newton method for the logistic regression

Data: $\mathbf{H}, \mathbf{y}$ training data
Result: $\mathbf{x}$ model parameters
while *not converged* **do**
  $\mathbf{p} \leftarrow \dfrac{\exp^{\mathbf{Hx}}}{1+\exp^{\mathbf{Hx}}}$ — a component wise division
  $W_{ii} \leftarrow p_i(1-p_i), \quad i=1,n$ — defined equation (16)
  $\mathbf{z} \leftarrow \mathbf{Hx} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})$
  $\mathbf{x} \leftarrow (\mathbf{H}^t\mathbf{WH})^{-1}\mathbf{H}^t\mathbf{W}\,\mathbf{z}$
end

The value of $\mathbf{x}^{(k+1)}$ can also be obtained from $\mathbf{x}^{(k)}$ solving a reweighted least square problem [for more details see Hastie et al., 2005]. Indeed,

$$\begin{aligned}\mathbf{x}^{(k)} - (\mathbf{H}^t\mathbf{WH})^{-1}\mathbf{H}^t(\mathbf{p}-\mathbf{y}) &= (\mathbf{H}^t\mathbf{WH})^{-1}\big((\mathbf{H}^t\mathbf{WH})\mathbf{x}^{(k)} - \mathbf{H}^t(\mathbf{p}-\mathbf{y})\big)\\ &= (\mathbf{H}^t\mathbf{WH})^{-1}\mathbf{H}^t\mathbf{W}\mathbf{z},\end{aligned}$$

with $\mathbf{z} = \mathbf{Hx}^{(k)} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})$. Algorithm 1, where the division is considered elementwise, is implementing this solution.

4 Constrained convex and differentiable optimization

4.1 The theory of constrained convex and differentiable optimization

In the general case of constrained optimization (problem $(\mathcal{P})$ in (5)), necessary and sufficient optimality conditions are given through the Karush, Kuhn and Tucker (KKT) conditions defined as follows:

Definition 9 (Karush, Kuhn and Tucker (KKT) conditions)**.** *Vectors* ($\mathbf{x} \in \mathbb{R}^n, \boldsymbol{\lambda} \in \mathbb{R}^p, \boldsymbol{\mu} \in \mathbb{R}^q$) *are verifying the KKT condition of problem* $\mathcal{P}$ *if:*

stationarity	$\boldsymbol{\nabla} F(\mathbf{x}) + \sum_{j=1}^{p} \lambda_j \boldsymbol{\nabla} h_j(\mathbf{x}) + \sum_{i=1}^{q} \mu_i \boldsymbol{\nabla} g_i(\mathbf{x}) = \mathbf{0}$	
primal admissibility	$h_j(\mathbf{x}) = 0$	$j = 1, \ldots, p$
	$g_i(\mathbf{x}) \leq 0$	$i = 1, \ldots, q$
dual admissibility	$\mu_i \geq 0$	$i = 1, \ldots, q$
complementarity	$\mu_i g_i(\mathbf{x}) = 0$	$i = 1, \ldots, q.$

λ_j *and* μ_i *are called the Lagrange multipliers of problem* $\mathcal{P}$*.*

Variables $\mathbf{x}$ are also called primal variables and $\boldsymbol{\lambda}, \boldsymbol{\mu}$ dual variables. The intuition behind this optimality condition again lies within Fermat's Theorem since optimality is obtained again when the gradient of an objective function is

canceled (stationarity). If one cannot find a solution that cancels $\boldsymbol{\nabla} F$ then one must find a solution where the constraints will cancel the gradient as illustrated in the following example.

Example 7 (Simple KKT conditions)**.** *The following 1-dimensional optimization problem*

$$\min_{x \geq 0} \quad \frac{1}{2}(x+1)^2$$

whose solution is obviously $x^\star = 0$ leads to the following KKT conditions

$$\begin{array}{ll} \textit{stationarity} & (x+1) - \mu = 0 \\ \textit{primal admissibility} & -x \leq 0 \\ \textit{dual admissibility} & \mu \geq 0 \\ \textit{complementarity} & \mu x = 0 \end{array}$$

we can see that at the optimality, x is on the positivity constraint, which means that the gradient has to be canceled by $\mu = 1$. The complementarity condition impose that only x or μ are active at the same time which means that $\mu \neq 0$ only if x is on the constraint.

Example 8 (Constrained least square)**.** *Consider the following particular case of problem* (2) *with a least square loss*

$$\left\{ \begin{array}{ll} \min\limits_{\mathbf{x} \in \mathbb{R}^n} & \frac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 \\ \textit{with} & 0 \leq x_i, \qquad i = 1, \ldots, n \end{array} \right.$$

The KKT of this problem are

$$\begin{array}{ll} \textit{stationarity} & \mathbf{H}^t(\mathbf{H}\mathbf{x} - \mathbf{y}) - \boldsymbol{\mu} = \mathbf{0} \\ \textit{primal admissibility} & -\mathbf{x} \leq \mathbf{0} \\ \textit{dual admissibility} & \boldsymbol{\mu} \geq \mathbf{0} \\ \textit{complementarity} & \mathbf{diag}(\boldsymbol{\mu})\mathbf{x} = \mathbf{0} \end{array}$$

Theorem 2. *[Theorem 12.1 in Nocedal and Wright, 2006] A vector $\mathbf{x}^\star$ is the solution of a convex problem $\mathcal{P}$, that is the global minima if there exists, under linear independence constraint qualification, Lagrange multipliers $\{\lambda_j^\star\}_{j=1,\ldots,p}, \{\mu_i^\star\}_{i=1,\ldots,q}$ such that $\left(x^\star, \{\lambda_j^\star\}_{j=1,\ldots,p}, \{\mu_i^\star\}_{i=1,\ldots,q}\right)$ fulfill the KKT conditions.*

In the non convex case, these conditions characterize a stationary point. To compute the stationary condition, it is handy to introduce the Lagrangian function associated with problem $\mathcal{P}$.

Definition 10. *The Lagrangian $\mathcal{L}$ of problem $\mathcal{P}$ is the following function:*

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = F(x) + \sum_{j=1}^{p} \lambda_j h_j(\mathbf{x}) + \sum_{i=1}^{q} \mu_i g_i(\mathbf{x}) \tag{17}$$

The Lagrangian facilitates the calculus of the stationarity condition of Definition 9 since it is given by $\boldsymbol{\nabla}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0}$. Also, the solution of the optimization problem is the Lagrangian saddle point and is given by

$$\max_{\boldsymbol{\lambda},\boldsymbol{\mu}} \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}).$$

Example 9 (Lagrangian of the constrained least square from Example 8)**.**

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) = \tfrac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 - \boldsymbol{\mu}^t\mathbf{x}$$

Example 10 (Lagrangian formulation of example 4)**.** *Consider the following convex constrained optimization problem with the notations of example* (4) *and a given* $k > 0$

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & L(\mathbf{x}) \\ \textit{with} & \Omega(\mathbf{x}) \leq k, \end{cases} \tag{18}$$

with both functions L *and* Ω *convex. The Lagrangian of this problem is*

$$\mathcal{L}(\mathbf{x}, \mu) = L(\mathbf{x}) + \mu\big(\Omega(\mathbf{x}) - k\big),$$

that is, for a given k *it exists a* μ *solution of the problem so that this problem is equivalent to solve* (4)*, called the Lagrangian formulation of problem* (18)*. Applying the same reasoning, the problem*

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \Omega(\mathbf{x}) \\ \textit{with} & L(\mathbf{x}) \leq \ell, \end{cases} \tag{19}$$

and the two other formulations (18) *and* (4) *are equivalent. Note that this equivalence is due to the convex nature of the problem.*

The Lagrange dual objective function Q is defined from the Lagrangian

$$\begin{aligned} Q(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \inf_{\mathbf{x}\in\mathbb{R}^n} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ &= \inf_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}) + \sum_{j=1}^{p} \lambda_j h_j(\mathbf{x}) + \sum_{i=1}^{q} \mu_i g_i(\mathbf{x}) \end{aligned}$$

and the dual problem defined below.

Definition 11 (Dual problem)**.** *The dual problem of* $\mathcal{P}$ *is*

$$\mathcal{D} = \begin{cases} \max\limits_{\boldsymbol{\lambda}\in\mathbb{R}^p, \boldsymbol{\mu}\in\mathbb{R}^q} & Q(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \textit{with} & \mu_j \geq 0, \qquad j = 1, \ldots, q \end{cases}$$

Example 11 (Constrained least square from Example 8)**.**

$$Q(\boldsymbol{\mu}) = \inf_{\mathbf{x}} \quad \tfrac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 - \boldsymbol{\mu}^t\mathbf{x}$$

The gradient of the Lagrangian w.r.t. $\mathbf{x}$ *is* $\mathbf{H}^t(\mathbf{H}\mathbf{x} - \mathbf{y}) - \boldsymbol{\mu}$ *which gives us* $\mathbf{x}^\star = (\mathbf{H}^t\mathbf{H})^{-1}(\boldsymbol{\mu} + \mathbf{H}^t\mathbf{y})$*. When injected in the Lagrangian, we can find that*

$$Q(\boldsymbol{\mu}) = -\tfrac{1}{2}(\mathbf{H}^t\mathbf{y} + \boldsymbol{\mu})(\mathbf{H}^t\mathbf{H})^{-1}(\mathbf{H}^t\mathbf{y} + \boldsymbol{\mu})$$

and the dual problem is

$$\left\{ \begin{array}{ll} \min\limits_{\boldsymbol{\mu}\in\mathbb{R}^q} & \frac{1}{2}\boldsymbol{\mu}^t(\mathbf{H}^t\mathbf{H})^{-1}\boldsymbol{\mu} + \mathbf{y}^t\mathbf{H}(\mathbf{H}^t\mathbf{H})^{-1}\boldsymbol{\mu} \\ \textit{with} & \mu_j \geq 0, \qquad j = 1, \ldots, q, \end{array} \right.$$

which is also a QP with positivity constraints. Note that thanks to the stationarity condition in example (8) $\boldsymbol{\mu} = \mathbf{H}^t(\mathbf{H}\mathbf{x} - \mathbf{y})$*, so that the dual can be also expressed using the primal variables as*

$$\left\{ \begin{array}{ll} \max\limits_{\mathbf{x}\in\mathbb{R}^n} & -\frac{1}{2}\mathbf{x}^t\mathbf{H}^t\mathbf{H}\mathbf{x} \\ \textit{with} & \mathbf{H}^t(\mathbf{H}\mathbf{x} - \mathbf{y}) \geq 0. \end{array} \right.$$

Theorem 3 (Duality gap, 12.12, 12.13 and 12.14 Nocedal & Wright pp 346)**.** *If* F, g *and* h *are convex and continuously differentiable, under some constraint qualification conditions the cost of the dual solution is the same as the cost of the primal solution.*

For any feasible point $\mathbf{x}$ we have $Q(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq F(\mathbf{x})$ so that $0 \leq F(\mathbf{x}) - Q(\boldsymbol{\lambda}, \boldsymbol{\mu})$. This difference between the primal and dual cost functions is called the duality gap and is always positive.

4.2 Advanced example: support vector data description

As an example of constrained optimization problem we propose to solve the minimum enclosing ball problem introduced in Tax and Duin [2004] as *support vector data description* (SVDD). The name support vector refers to the fact that the boundary between classes will lean on some specific vectors of the training data set. Given p points $\{\mathbf{h}_i \in \mathbb{R}^n, i = 1, \ldots, p\}$ this problem consists in finding the n dimensional ball of centre $\mathbf{c}$ with minimum radius R that contains all the points $\mathbf{h}_i$. The SVDD problem can be expressed as follows:

$$\left\{ \begin{array}{ll} \min\limits_{R\in\mathbb{R},\mathbf{c}\in\mathbb{R}^n} & R^2 \\ \text{with} & \|\mathbf{h}_i - \mathbf{c}\|^2 \leq R^2, \qquad i = 1, \ldots, p. \end{array} \right. \tag{20}$$

The associated Lagrangian is:

$$\mathcal{L}(\mathbf{c}, \rho, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{c}\|^2 - \rho - \sum_{i=1}^{p} \mu_i\big(\mathbf{c}^t\mathbf{h}_i - \rho - \frac{1}{2}\|\mathbf{h}_i\|^2\big),$$

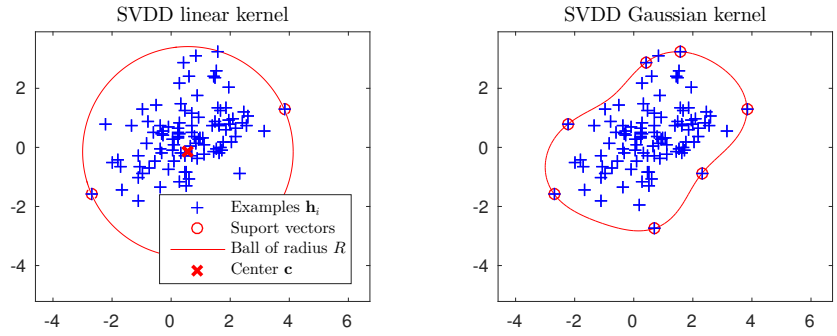

Figure 5: Illustration of SVDD for a linear kernel (left) and a Gaussian kernel (right) on the same dataset. The red curve show the minimum enclosing ball.

with the KKT conditions:

$$\begin{array}{lll}
\text{stationarity} & \mathbf{c} - \sum_{i=1}^{p} \mu_i \mathbf{h}_i = 0 & \\
& 1 - \sum_{i=1}^{p} \mu_i = 0 & \\
\text{primal admissibility} & \mathbf{c}^t \mathbf{h}_i \geq \rho + \frac{1}{2}\|\mathbf{h}_i\|^2 & i = 1, \ldots, p \\
\text{dual admissibility} & \mu_i \geq 0 & i = 1, \ldots, p \\
\text{complementarity} & \mu_i \big(\mathbf{c}^t \mathbf{h}_i - \rho - \frac{1}{2}\|\mathbf{h}_i\|^2\big) = 0 & i = 1, \ldots, p.
\end{array}$$

Complementarity tells us that there are two groups of points: the support vectors lying on the circle (for which $\|\mathbf{h}_i - \mathbf{c}\|^2 = R^2$) and the insiders for which $\mu_i = 0$. Stationarity brings the relationship so-called the representer theorem:

$$\mathbf{c} = \sum_{i=1}^{p} \mu_i \mathbf{h}_i. \tag{21}$$

Some tedious calculus bring us the dual formulation of the SVDD as the following QP with $\mathbf{G} = \mathbf{H}\mathbf{H}^t$ the Gram matrix of general term $G_{ij} = \mathbf{h}_i^t \mathbf{h}_j$,

$$\left\{ \begin{array}{lll}
\min\limits_{\boldsymbol{\mu} \in \mathbb{R}^p} & \boldsymbol{\mu}^t \mathbf{G} \boldsymbol{\mu} - \boldsymbol{\mu}^t \ \mathbf{diag}(\mathbf{G}) & \\
\text{with} & \boldsymbol{e}^t \boldsymbol{\mu} = 1 & \\
\text{and} & 0 \leq \mu_i & i = 1, \ldots, n.
\end{array} \right. \tag{22}$$

Pros and cons of these primal and dual formulations are summarized table 1. Figure 5 (left) illustrates a 2d SVDD.

In Astronomy, this problem is for instance encountered in minimax detection of spectral profiles [Suleiman et al., 2014]. In this application, we are given a library of spectral profiles $\{\mathbf{h}_i \in \mathbb{R}^n, i = 1, \ldots, p\}$. One wishes to design a profile that has the largest minimal correlation with all profiles. This optimal profile is precisely the center $\mathbf{c}$ of the minimum enclosing ball.

Table 1: Pros and cons of the SVDD primal and dual formulation

Primal (20)	Dual (22)
$n+1$ unknown	p unknown
p constraints	n box constraints
can be recast as a QP	build G the $p \times p$ pairwise influence Gram matrix
perfect when $n < p$	to be used when $n > p$

SVDD (20) allows to model a set of observations by a circle surrounding the data. Two pitfalls prevents SVDD to model properly a real set of observations: it tolerates no error and it is limited to circle. We will see now how to adapt the initial SVDD model to address these two issues respectively by introducing slack variables and kernel.

It is possible to relax model (20) and deal with potential errors by introducing slack variables $\xi_i, i = 1, \ldots, p$ associated with each observation defined as

$$\text{for all } \mathbf{h}_i \qquad \begin{cases} \text{no error:} & \|\mathbf{h}_i - \mathbf{c}\|^2 \leq R^2 \Rightarrow \quad \xi_i = 0 \\ \text{error:} & \|\mathbf{h}_i - \mathbf{c}\|^2 > R^2 \Rightarrow \quad \xi_i = \|\mathbf{h}_i - \mathbf{c}\|^2 - R^2. \end{cases}$$

Introducing these slack variable in the initial SVDD setting (20) is generalized by, for a given parameter $C > 0$

$$\begin{cases} \min\limits_{R \in \mathbb{R}, \mathbf{c} \in \mathbb{R}^n, \boldsymbol{\xi} \in \mathbb{R}^p} & R^2 + C \sum\limits_{i=1}^{n} \xi_i & \\ \text{with} & \|\mathbf{h}_i - \mathbf{c}\|^2 \leq R^2 + \xi_i & i = 1, \ldots, p \\ & 0 \leq \xi_i & i = 1, \ldots, p. \end{cases} \tag{23}$$

This problem amounts to the initial SVDD (20) when $C \to \infty$. The generalized SVDD is associated with the dual

$$\begin{cases} \min\limits_{\boldsymbol{\mu} \in \mathbb{R}^p} & \boldsymbol{\mu}^t \mathbf{G} \boldsymbol{\mu} - \boldsymbol{\mu}^t \ \mathbf{diag}(\mathbf{G}) & \\ \text{with} & \boldsymbol{e}^t \boldsymbol{\mu} = 1 & \\ \text{and} & 0 \leq \mu_i \leq C & i = 1, \ldots, p. \end{cases} \tag{24}$$

Note that the introduction of the slack variables only adds a box constraint on the dual.

An efficient way to introduce non linearities to go beyond the SVDD circular model, consists in using kernels functions as features. The results of kernelized SVDD is illustrated in Figure 8 (right) where we can see that, thanks to the kernel, the red circle model is distorted to provide a better fit to the data. A kernel in this framework is a positive function of two variables. Popular kernels are the Gaussian kernel k_g with bandwidth $\sigma > 0$ and k_p the polynomial kernel of order d:

$$k_g(\mathbf{h}, \mathbf{h}') = \exp\left(\frac{-\|\mathbf{h} - \mathbf{h}'\|^2}{\sigma}\right), \qquad k_p(\mathbf{h}, \mathbf{h}') = (1 + \mathbf{h}^t \mathbf{h}')^d. \tag{25}$$

Associated with each kernel, a norm $\|\cdot\|_{\mathcal{H}}$ can be defined and used to define the following kernelized version of the SVDD [for more details on kernel machines see Smola and Schölkopf, 1998]:

$$\left\{\begin{array}{ll} \min\limits_{R\in\mathbb{R},\mathbf{c}\in\mathcal{H},\boldsymbol{\xi}\in\mathbb{R}^p} & R^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{with} & \|k(\mathbf{h},\mathbf{h}_i) - \mathbf{c}(\mathbf{h})\|^2_{\mathcal{H}} \leq R^2 + \xi_i \qquad i = 1,\ldots,p \\ & 0 \leq \xi_i \qquad\qquad\qquad\qquad\quad i = 1,\ldots,p. \end{array}\right. \tag{26}$$

Kernelized SVDD (26) are almost the same as SVDD (24). The only difference is that the notion of minimum enclosing ball is taken according to the norm $\|\cdot\|_{\mathcal{H}}$ associated with the kernel. Nevertheless, it turns out that the introduction of this new norm does not change much of the model since the dual of this problem remains the same as (24) but with $G_{ij} = k(\mathbf{h}_i, \mathbf{h}_j)$. In that case, the primal problem is of infinite dimension and thus intractable while the dual problem is still of dimension p. After solving the dual, the dual variables $\mu_i, i = 1,\ldots,p$ and ρ are known, the representer theorem 21 becomes $c(\mathbf{h}) = \sum_{i=1}^{n}\mu_i k(\mathbf{h},\mathbf{h}_i)$ and the function defining the enclosing ball of the SVDD is given by

$$\begin{aligned} \|k(\mathbf{h},.) - c(.)\|^2_{\mathcal{H}} - R^2 &= \|k(\mathbf{h},.)\|^2_{\mathcal{H}} - 2\langle k(\mathbf{h},.), c(.)\rangle_{\mathcal{H}} + \|c(.)\|^2_{\mathcal{H}} - R^2 \\ &= -2c(\mathbf{h}) + k(\mathbf{h},\mathbf{h}) - \rho \\ &= -2\sum_{i=1}^{n}\mu_i k(\mathbf{h},\mathbf{h}_i) + k(\mathbf{h},\mathbf{h}) - \rho. \end{aligned}$$

5 Nondifferentiable unconstrained optimization

5.1 The proximal algorithm for non-differentiable optimization

There exists different ways of dealing with non differentiability. We can use subdifferentials, convex relaxations or proximal algorithms [Combettes and Pesquet, 2011, Parikh and Boyd, 2014]. In this section we investigate the case where the objective function is a composite function $F(\mathbf{x}) = L(\mathbf{x}) + \lambda\Omega(\mathbf{x})$ as in problem (4) with L a differentiable loss and Ω a non-differentiable penalty function. This kind of problem is common in signal processing and statistical learning. It is of particular interest for sparsity promoting optimization *i.e.* when we want the solution vector $\mathbf{x}$ to have few non-zero components. The introduction has provided several examples in radio image reconstruction that use sparsity. See Bach et al. [2011] for more details about sparsity and group sparsity.

In order to solve the non-differentiable optimization problem, proximal methods rely (again) on the minimization of a simple majorization of the function. If we suppose that the function $L(\mathbf{x})$ is Lipschitz gradient then at iteration k we have

$$F(\mathbf{x}) \leq L(\mathbf{x}^{(k)}) + \boldsymbol{\nabla} L(\mathbf{x}^{(k)})^t(\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2\rho}\|\mathbf{x} - \mathbf{x}^{(k)}\|^2 + \lambda\Omega(\mathbf{x}), \tag{27}$$

when $\rho \leq \frac{1}{L_L}$ with L_L the Lipschitz constant of function $L(\mathbf{x})$. This majorization can be minimized easily when the penalty function $\Omega(\mathbf{x})$ is simple (separable for instance). Minimizing the majorization above can be reformulated as

$$\min_{\mathbf{x}} \quad \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda\rho\Omega(\mathbf{x}),$$

with $\mathbf{u} = \mathbf{x}^{(k)} - \mu\boldsymbol{\nabla} L(\mathbf{x}^{(k)})$. The expression above is a proximity operator as defined below.

Definition 12 (Proximity operator)**.** *The Proximity operator of a function Ω is:*

$$\begin{aligned}\mathbf{prox}_{\Omega} : \mathbb{R}^n &\longrightarrow \mathbb{R}^n \\ \mathbf{x} &\longmapsto \mathbf{prox}_{\Omega}(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathbb{R}^n} \ \Omega(\mathbf{u}) + \tfrac{1}{2}\|\mathbf{u}-\mathbf{x}\|^2.\end{aligned}$$

As discussed above this operator is a key step when minimizing the composite function, which leads to the following optimization algorithm.

Definition 13 (Proximal gradient descent)**.** *A general setting to solve the minimization problem (4) consists in the following iterations*

$$\mathbf{x}^{k+1} = \mathbf{prox}_{\rho^{(k)}\lambda\Omega}\big(\mathbf{x}^k - \rho^{(k)}\boldsymbol{\nabla} L(\mathbf{x}^k)\big).$$

This approach is also known as proximal splitting and Forward Backward splitting in the signal processing community. It can be generalized to an arbitrary number of composite functions [Combettes and Pesquet, 2011].

The gradient step ρ can be fixed *a priori* from the Lipschitz constant or found at each iteration to ensure both convergence speed and reduction of the cost function. The efficiency in terms of convergence speed of the proximal gradient method depends on the penalty function $\Omega(\mathbf{x})$. In particular when $\Omega(\mathbf{x})$ is componentwise separable, as it is Equation (8), the proximity operator is easy to compute. The following example shows several simple proximity operators for different componentwise separable penalty functions and Figure 6 plot the 1-dimensional operator $\omega(x)$ with $\lambda = 1$.

Example 12 (Common proximity operators)**.**

$\Omega(\mathbf{x}) = 0$	$\mathbf{prox}_{\Omega}(\mathbf{x}) = \mathbf{x}$	*identity*
$\Omega(\mathbf{x}) = \lambda\|\mathbf{x}\|_2^2$	$\mathbf{prox}_{\Omega}(\mathbf{x}) = \frac{1}{1+\lambda}\mathbf{x}$	*scaling*
$\Omega(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$	$\mathbf{prox}_{\Omega}(\mathbf{x}) = \mathit{sign}(\mathbf{x})\max(0, \lvert\mathbf{x}\rvert - \lambda)$	*soft shrinkage*
$\Omega(\mathbf{x}) = \lambda\|\mathbf{x}\|_{1/2}^{1/2}$	*[Xu et al., 2012, Equation 11]*	*bridge or power family*
$\Omega(\mathbf{x}) = \mathbb{1}_C(\mathbf{x})$	$\mathbf{prox}_{\Omega}(\mathbf{x}) = \underset{\mathbf{u}\in C}{\mathit{argmin}} \ \frac{1}{2}\|\mathbf{u}-\mathbf{x}\|^2$	*hard shrinkage projection.*

Note that the last function in Example 12 is an indicator function that is $+\infty$ for all $\mathbf{x}$ outside the set C. This illustrates the fact that one can use this framework even for constrained optimization, in this case the operator is a projection and the algorithm reverts to projected gradient descent.

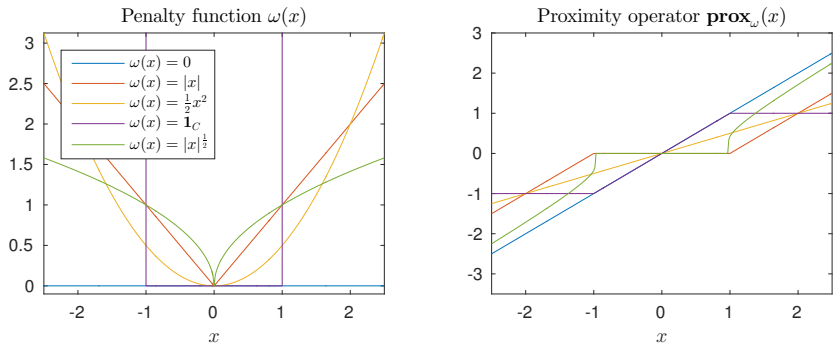

Figure 6: Illustration of several penalty functions (left) and their proximity operator (right). The set C for the indicator function is set to $[-1, 1]$.

The algorithm defined in Definition 13 has be shown to converge to the global minimum of the objective function under some mild assumption on the step ρ. In practice in order to have a better convergence speed some have recommended to use the Barzilai-Borwein rule that aim at estimating locally the curvature of the function [Barzilai and Borwein, 1988, Wright et al., 2009, Gong et al., 2013]. Note that there exists a family of accelerated proximal gradients that have been introduced by Nesterov et al. [2007]. Those accelerated methods use an inertial descent direction and can reach a given value of the objective function in $\sqrt{n}$ iterations instead of n for classical gradient descent [Beck and Teboulle, 2009]. There has also been recently a vivid interest in primal-dual approaches that can be seen as proximal methods but rely on both primal and dual problems to find quickly a solution for convex optimization, see [Komodakis and Pesquet, 2015] for a good introduction.

Finally we discuss the proximal methods when the functions are not convex. Theoretical results have shown, under some mild conditions on the objective function, that the proximal algorithm converges to a stationary point of the optimization problem, that is in this case a local optimum [Attouch et al., 2010]. In practice, majorization (27) do not rely on the convexity of the problem and the proximal algorithm leads to a decrease of the cost at each iteration [Gong et al., 2013]. This approach is of particular interest when the penalty $\Omega(\mathbf{x})$ can be efficiently computed, for instance in the MCP case presented below section 5.2.2 or when the penalty function is the ℓ_p pseudo-norm with $p = \frac{1}{2}$ (proximity operator illustrated in Figure 6).

5.2 Advanced examples: sparse estimation

5.2.1 The sparse least squares

An example of convex nondifferential optimization problem is sparse least square also known as the Lasso [Tibshirani, 1996]. It aims at minimizing the sum of the square error and a L_1 norm penalization term that is, given a design matrix

$\mathbf{H}$, a response vector $\mathbf{y}$ and a parameter $\mu > 0$

$$\min_{\mathbf{x}\in\mathbb{R}^n} \ \tfrac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|^2 + \mu\sum_{i=1}^{n}|\mathbf{x}_i|. \tag{28}$$

This objective function is a composite function with $L(\mathbf{x}) = \frac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|^2$ and $\Omega(\mathbf{x}) = \sum_{i=1}^{n}|\mathbf{x}_i|$ the L_1 norm of the unknown variables. Using the second-order approximation (27) at each iteration, the associated proximal operator can be written as the sum of independent componentwise terms since

$$\begin{aligned}\mathbf{prox}_{\Omega}(\mathbf{x}) &= \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \sum_{i=1}^{n}|u_i| + \tfrac{1}{2}\|\mathbf{u}-\mathbf{x}\|^2 \\ &= \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \sum_{i=1}^{n}\big(|u_i| + \tfrac{1}{2}(u_i-x_i)^2\big).\end{aligned}$$

Because of the absolute value this is not differentiable. Its subdifferential is (see example 2)

$$\partial(|u_i| + \tfrac{1}{2}(u_i-x_i)^2) = \begin{cases} \operatorname{sign}(u_i) & + u_i - x_i & & \text{if } u_i \neq 0 \\ g & + u_i - x_i & \text{with } -1 \leq g \leq 1 & \text{if } u_i = 0, \end{cases}$$

so that

$$0 \in \partial(|u_i| + \tfrac{1}{2}(u_i-x_i)^2) \quad \Leftrightarrow \quad u_i = \begin{cases} \operatorname{sign}(u_i)(|x_i|-1) & \text{if } |x_i| > 1 \\ 0 & \text{if } |x_i| \leq 1. \end{cases}$$

Putting all that together with definition 13 lead to the proximal gradient descent algorithm for the Lasso given in Algorithm 3.

The optimality condition given above illustrates why L_1 regularization promote sparsity. Indeed, when the function to be optimized is non differentiable in zero, the condition for having a zero component u_i is not an equality but an inclusion, that is the possibility to cancel the gradient with any vector from a set. This inclusion is easier to achieve than an exact equality since it only requires that a component x_i is smaller than a threshold for the solution u_i to be exactly equal to zero as illustrated Figure 6 (right). This also illustrates that each iteration is the result of a proximal operator which often leads to sparse vectors. This can dramatically increase the speed of computation of the gradient (in particular of $\mathbf{H}\mathbf{x}$) when using sparse vector encoding.

5.2.2 Trading convexity for generalization: the sparse logistic regression

Another example of nondifferential but this time non convex optimization problem is the one associated with the non convex minimax concave penalty (MCP) penalized logistic regression. The idea of the MCP logistic regression consists in minimizing together with the logistic loss defined in (14) a MCP penalty term

Algorithm 2 The proximal gradient descent algorithm for the Lasso

Data: $\mathbf{H}, \mathbf{y}$ training data
Result: $\mathbf{x}, \mathbf{y}_{pred}$
$\rho \leftarrow 1/\|\mathbf{H}^t\mathbf{H}\|$ stepsize initialization
while *not converged* **do**
  $\mathbf{x} \leftarrow \mathbf{x} - \rho\mathbf{H}^t(\mathbf{H}\mathbf{x} - \mathbf{y})$ gradient forward step
  $\mathbf{x} \leftarrow \text{sign}(\mathbf{x})\max(0, |\mathbf{x}| - \rho\mu)$ proximal backward step
end

promoting generalization performances as well as sparsity through variable selection. This MCP penalty can be seen as an improvement over the Lasso L_1 norm penalization used equation in (28), that can cause *significant bias toward 0 for large regression coefficients*. On the opposite, it can be shown that the *MCP regression model has the so-called oracle property, meaning that, in the asymptotic sense, it performs as well as if the analyst had known in advance which coefficients were zero and which were nonzero* [Breheny and Huang, 2011]. The resulting optimization problem has the general form of (4) and can be written as

$$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^{p}\Big(-y_i(\mathbf{H}\mathbf{x})_i + \log\big(1+\exp(\mathbf{H}\mathbf{x})_i\big)\Big) + \sum_{i=1}^{n}\Omega_{\mu,\gamma}(|\mathbf{x}_i|), \tag{29}$$

where $\Omega_{\mu,\gamma}$ is the non convex MCP function defined by, for a given couple $(\lambda \geq 0, \gamma \geq 1)$ of hyper parameters:

$$\Omega_{\mu,\gamma}(t) = \begin{cases} \mu t - \dfrac{t^2}{2\gamma} & \text{if } t \leq \gamma\mu \\ \dfrac{\gamma\mu^2}{2} & \text{else.} \end{cases}$$

Parameter μ controls the tradeoff between the loss function and penalty, while parameter γ controls the shape of the penalty as shown figure 7. The solution of the MCP logistic regression has nice statistical properties but the resulting optimization is challenging due to the non-convexity and non-differentiability of the penalty term.

To deal with these difficulties, the MCP penalty can be decomposed as the difference of two functions so that problem (29) can be written as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \underbrace{\sum_{i=1}^{p}\big(-y_i(\mathbf{H}\mathbf{x})_i + \log\big(1+\exp((\mathbf{H}\mathbf{x})_i)\big)\big) - \mu\sum_{j=1}^{n} h_{\mu,\gamma}(|x_j|)}_{F_m(\mathbf{x})} + \mu\|\mathbf{x}\|_1, \tag{30}$$

with

$$h_{\mu,\gamma}(t) = \left\{ \frac{t^2}{2\gamma\mu}\mathbb{1}_{\{t\leq\gamma\mu\}} + \left(t - \frac{\gamma\mu}{2}\right)\mathbb{1}_{\{t>\gamma\mu\}} \right\},$$

the Huber penalty function with parameter $\gamma\mu$, illustrated figure 7.

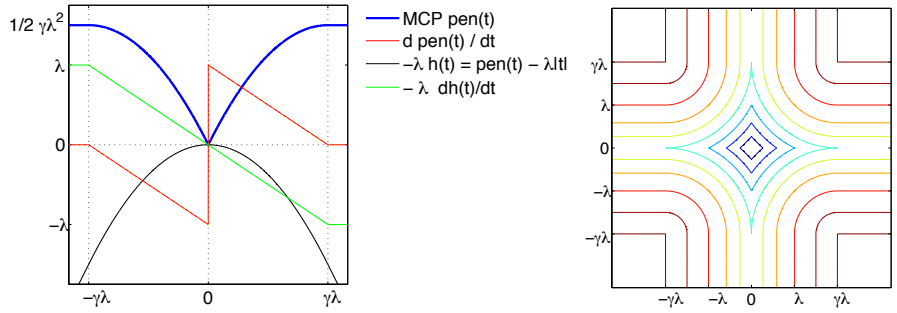

Figure 7: Left: Function $\Omega_{\mu,\gamma}(t)$ (blue) for $\mu = 1$ and $\gamma = 3$, its derivative (red), the associated inverse of the Huber loss (black) and its derivative (green). Right: the MCP penalty level set for $\mu = \gamma = 2$.

In this optimization problem the loss function is the sum of $F_m(\mathbf{x})$ a non convex but differentiable function with a L_1 norm convex but nondifferentiable. One way to handle this non differentiability is to apply the proximal projection of the L_1 norm on the gradient of F_m. The gradient of F_m is (see the green curve figure 7)

$$\boldsymbol{\nabla}_\beta F_m(\mathbf{x}) = \mathbf{H}^t(\mathbf{p} - \mathbf{y}) - \begin{cases} \text{sign}(x_j)\mu & \text{if } |x_j| > \mu\gamma \\ \dfrac{x_j}{\gamma} & \text{else.} \end{cases}$$

The proximal operator of the L_1 penalty function is

$$\mathbf{prox}(u) \;=\; \begin{cases} 0 & \text{if } |u| \leq \mu \\ \text{sign}(u)(|u| - \mu) & \text{else.} \end{cases}$$

The whole proximal gradient procedure is summarized algorithm 3. For a well chosen stepsize $\rho \leq \frac{1}{\sigma_M^2}$, σ_M being the largest singula value of the design matrix $\mathbf{H}$, this algorithm converges towards a local minima of problem (29). Indeed, due to the nonconvexity of the problem, global convergence cannot be proven while using a proximal algorithm.

6 Optimization in practice

6.1 Classification of RR Lyrae

An example where optimization problems occur in astronomy is when dealing with supervised binary classification. To illustrate the problem, we consider a RR Lyrae dataset taken from astroML [datasets based on Stripe 82 data in Ivezić et al., 2014]. This sample contains 92658 stars flagged as nonvariable and 483 RR-Lyrae observed in $n = 4$ dimensions namely the $u - g$, $g - r$, $r - i$ and $i - z$ colors build upon the $u, g, r, i, z-$band photometry [for details see Ivezić

Algorithm 3 The L_1 proximal algorithm for the MCP penalized logistic regression

Data: $\mathbf{H}, \mathbf{y}$ training data, $\mathbf{H}_{test}$, test data
Result: $\mathbf{x}, \mathbf{y}_{pred}$
$\rho \leftarrow 1/\|\mathbf{H}^t\mathbf{H}\|$ stepsize initialization
while *not converged* **do**
  $\mathbf{p} \leftarrow \dfrac{\exp^{\mathbf{Hx}}}{1+\exp^{\mathbf{Hx}}}$ element-by-element division
  $\mathbf{x} \leftarrow \mathbf{x} - \rho\big(\mathbf{H}^t(\mathbf{p}-\mathbf{y}) - \text{sign}(\mathbf{x})\min(\mu, \frac{|\mathbf{x}|}{\gamma})\big)$
  $\mathbf{x} \leftarrow \text{sign}(\mathbf{x})\max(0, |\mathbf{x}| - \rho\mu)$
end

et al., 2005]. The task here is to design a classifier capable of predicting whether or not a new four dimensional vector observation comes from the observation of a *nonvariable star* or a *RR-Lyrae*. Specific difficulties of this task are the volume and the unbalanced structure of the training data together with the nonlinear nature of the problem (illustrated figure 8).

The unbalanced nature of the data suggest to gauge the quality of a learned model using the completeness, contamination and F_1 measures instead of the classical classification error rate. Indeed, classifying all data as a background object, *i.e. star*, would lead to a seemingly good classification error, 0.5% in this case, but this is of no practical interest. Instead, the completeness, contamination and F_1 measures are relevant because they do not take into account the fraction of background well classified as background, also called the true negative rate. The completeness is defined as the fraction of point classified as RR-Lyrae instances that are relevant, while efficiency is the fraction of relevant instances that are retrieved. More formally, if TP denotes the number of true positives (well classified RR-Lyrae) FP the number of false positives (the number of stars classified as RR-Lyrae) and FN the number of false negatives (the number of RR-Lyrae classified as stars), these measures are defined by

$$\text{completeness} = \frac{TP}{TP+FN} \qquad \text{efficiency} = \frac{TP}{TP+FP}.$$

In machine learning, these two terms are referred as precision and recall. A popular quality measure that combine efficiency and completeness is the F_1 score, defined as their harmonic mean:

$$F_1 \;=\; 2\,\frac{\text{completeness} \;\times\; \text{efficency}}{\text{completeness} + \text{efficency}}.$$

Given these measures, we propose to solve the supervised classification task with the use of the logistic regression together with kernels to deal with non linearities. However, the excessive amount of available data and its unbalanced nature require a reduction of the dataset. To this end, two mechanisms are

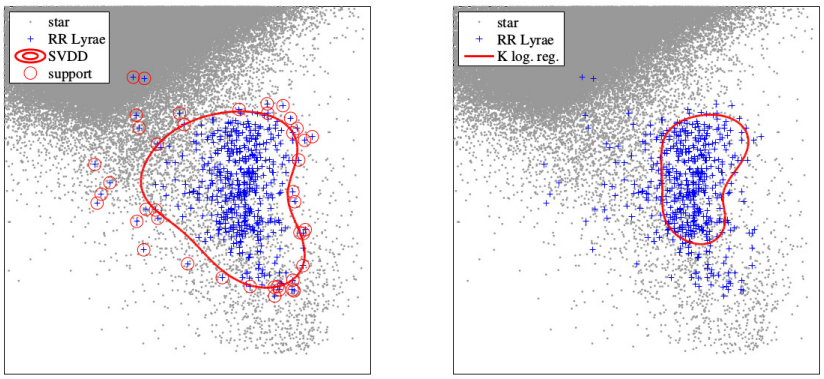

Figure 8: Kernelized SVDD (right) and kernelized logistic regression (left) on two dimensional Stripe 82 data, trained only with RR-Lyrae using a gaussian kernel with $C = 1/80$ and $\sigma = 4/3$ on centered and reduced data.

introduced: preprocessing to reduce the training set through eliminating almost surely unrelevant data from the too large *star class* and the use of a sparsity penalty term such as the MCP introduced in (29).

Finally, post processing is used to improve the resulting model by removing possible bias. To met all these requirements we propose the following three-step procedure:

1. pre processing to reduce the training set through eliminating unrelevant data,
2. classification with sparse kernelized logistic regression,
3. post processing to polish the resulting model.

The preprocessing is performed by using the kernelized version of SVDD QP program (24) trained on the 483 data points labeled RR-Lyrae. As often, a Gaussian kernel described in equation (25) is used with a bandwith of 4/3 and a $C = 1/80$. The results in two dimensions are illustrated figure 8 on the left side. Then, the resulting SVDD model is used on the 92658 background objects to select a fraction (2000) of nonvariable stars likely to be close to the decision function.

This selected data is then used in a second phase to train a kernelized version of the MCP penalized version of the logistic regression (algorithm 3 with $\mu = 2$ and $\gamma = 4$) that could not have processed the whole dataset. The results in two dimensions, illustrated on the right side of figure 8, show a tighter decision frontier (red curve). This method provides interesting results together with a new list of the data point selected to build the classification rule.

It turns out that the resulting model can be improved by a post process consisting on the training of a non penalized kernel logistic regression (algorithm 1) on the selected data that is 323 points in our case. The whole procedure is summarized algorithm 4.

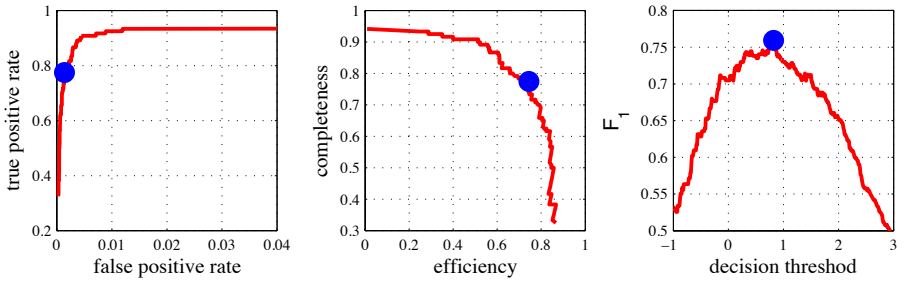

Figure 9: ROC curve (left), completeness-efficiency curve (center) and F_1 (left) for the four color RR Lyrae data using the classification algorithm 4. The kernelized logistic regression achieves , for the blue spot, a completeness of 0.77 for an efficency of 0.75 leading to a F_1 score of 0.76.

Algorithm 4 Classification algorithm and associated optimization methods

Data: $H, \mathbf{y}$ training data, H_{test}, test data
Result: $\mathbf{x}, \mathbf{y}_{pred}$
$\mu \leftarrow \text{KSVDD}(H_{RR-Lyrae}, y_{RR-Lyrae})$, QP in the dual
$pos \leftarrow \text{select}(H_{star}, \mu)$,
$pos \leftarrow \text{Sparse\_KLogisticRegression}(H(pos, :), y)$, proximal gradient
$\mathbf{x} \leftarrow \text{KLogisticRegression}(H(pos, :), y)$, Newton
$\mathbf{y}_{pred} \leftarrow H_{test}\mathbf{x}$

The global performances of the proposed approach are presented figure 9 following the same protocol as astroML[6]. Our best model, represented by a blue spot figure 9, reach a maximum F_1 score of 0.76 improving on 0.72 the best F_1 reported in astroML by using a Gaussian mixture-model. In this figure, three plots are proposed to visualize the performances of our method. On the left, the receiver operating characteristic, or ROC curve represents the true positive rate (TP) against the false positive rate (FP) at various decision threshold settings (see equation (15) for precision on the use of the threshold). Note that unlike most of the results reported in astroML [Ivezić et al., 2014, figure 9.17 page 396] our method fail at reaching a TP of one. However, results on the completeness vs. efficiency plot (in the center panel) indicates that our approach clearly outperforms results reported in astroML in the interesting zone of compromise, around the blue spot. The third plot on the right reports the sensitivity of the F_1 measure to the decision threshold.

In front of such a real problem with real data, the practitioner always asks himself about the classifier he should use. Very often the answer will to combine methods, including pre and post processing.

[6] `http://www.astroml.org/book_figures/chapter9/fig_ROC_curve.html`

6.2 MUSE spectrum denoising

In this section we apply non-differentiable optimization to spectrum denoising. To this end we propose to use sparse least square as already proposed in Bourguignon et al. [2012, 2010]. To keep this application example simple, we focus below on 1-dimensional spectra. In reality, MUSE instrument has a 3D PSF that smoothes out the spectrum and makes the reconstruction more difficult (and computationally intensive) than in the 1D case, as discussed in the references above.

The illustration below uses simulated but realistic data of MUSE. These data were used by MUSE consortium for various tests before MUSE was operational and and mimic very accurately real data like, for instance, in Bacon, R. et al. [2015]. The noisy observation is obtained by adding Gaussian noise with a magnitude leading to a 20dB signal to noise ratio (SNR). MUSE spectra cover 3600 wavelengths over an interval from 465 to 930 nm (visible light). The spectra used in this experiment are shown in the top part of Figure 10 along with their noisy observations.

In this application we will use a simple version of the optimization problem (3). We choose to use a ℓ_2 as data fitting term, which is commonly used due to its simplicity and the fact that its minimization corresponds to likelihood maximization for a signal corrupted with additive Gaussian noise. The signal is estimated by minimizing

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 + \lambda \sum_k \omega(|x_k|), \tag{31}$$

where $\mathbf{D}$ is a dictionary of elementary spectral shapes elements and $\omega(\bullet) = \bullet$ for the ℓ_1 regularization and $\omega(\bullet) = (\bullet)^p$ for the ℓ_p regularization with $p = \frac{1}{2}$. As discussed in section 5.2.1 the regularization term is non-differentiable in 0 and will promote sparsity in $\mathbf{x}$. We use a proximal gradient[7] descent as discussed in section 5 to solve problem (31). In this application we use an over-complete dictionary of 27683 elements which is larger than the size of $\mathbf{y}$. The dictionary is created similarly as in Bourguignon et al. [2010] and consists in a mixture of impulses, steps and low frequency variations. Sparsity here comes to the rescue because limiting the number of activated dictionary elements should allow a reconstruction of the spectrum that has a structure close to the true spectrum. This will always be true if the dictionnary is adapted to the considered signals, and this is the case here because the dictionary was specifically designed for astrophysical spectra. In contrast, the noise is essentially unstructured and would require a large number of dictionary elements to be reconstructed. Consequently, the noise is 'filtered-out' by the sparse synthesis process.

The reconstructed spectra $\hat{\mathbf{y}} = \mathbf{D}\hat{\mathbf{x}}$ (with $\hat{\mathbf{x}}$ the solution of the optimisation problem) for both ℓ_1 (middle) and ℓ_p (bottom) regularization are given in the four lower panels of Figure 10. For each approach we report, for the best

[7]The Octave/Matlab optimization toolbox is available at `https://github.com/rflamary/nonconvex-optimization`

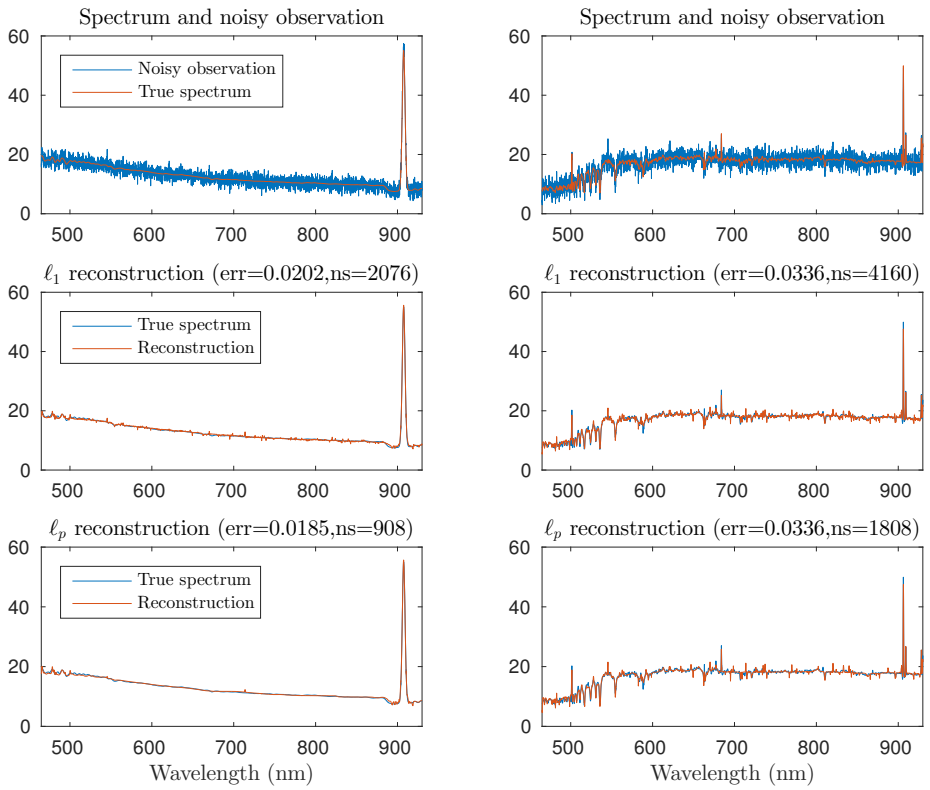

Figure 10: Example or spectrum denoising for two MUSE simulated spectra. (top) noisy observations and true spectrum, (middle) reconstruction using the ℓ_1 regularization (bottom) reconstruction using the ℓ_p regularization. For each approach we report the relative reconstruction error err and the number of selected atoms ns.

value of λ in (31), the relative error computed from the euclidean norm of the reconstruction error divided by the norm of the true spectrum. Note that both regularizations lead to similar relative errors but the ℓ_p regularization needs a lower number of dictionary elements ns to achieve the same performance. The ℓ_p $(0 \leq p < 1)$ regularization is indeed more aggressive in term of sparsity than ℓ_1 (as illustrated in Figure 6, right panel, compare green and red curves), but its shrinkage to zero is less strong as the magnitude of the components increase. In contrast, the ℓ_1 shrinkage is always the same. For instance, we can see in Figure 6 that a component of $x = 2$ will be shrinked to 1 with ℓ_1 penalization and to ≈ 1.7 with ℓ_p. Note that as discussed in Bourguignon et al. [2010] one can also perform a non regularized least square estimation on the components selected by ℓ_1 to diminish its bias.

The results reported in Figure 10 corresponds to the best performance for each regularization parameter. In order to find the best value for the parameter

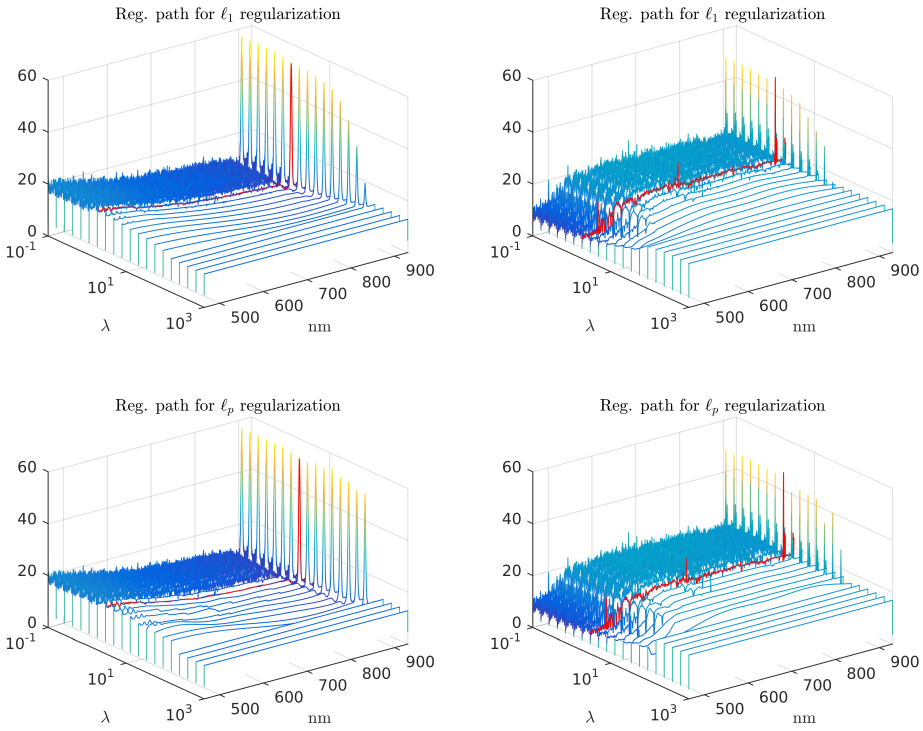

Figure 11: Regularization path for the two simulated spectra for ℓ_1 (top) and ℓ_p (bottom) regularization. The true spectrum is reported in red superposed with the reconstruction with the best reconstruction error.

λ we computed what is called an approximate regularization path, *i.e.* the solution of the optimization problem for a number of values of λ logarithmically sampled from 10^{-1} to 10^{-3}. The restored spectra for several vaules of λ are reported in Figures 11 for both spectra and regularizations. For all methods one can see that for very large regularization parameters the spectrum is represented only by its average value and progressively acquires a more complex shape as λ decreases. When λ is small, one can also see for both approaches that the reconstructed spectrum becomes noisy because the noise is also reconstructed. Finally, we can see in Figure 11 that the magnitude of the reconstructed spectrum changes abruptly along λ with ℓ_p regularization whereas the variation is more smooth with ℓ_1. Also, ℓ_1 regularization requires a smaller value of λ (and more selected features) at its optimal reconstruction error than ℓ_p. Both effects are related to their different types of shrinkage, as discussed above.

7 Conclusion

This article is only a short introduction to optimization and only scratches the surface of this very active field. At the same time, as pointed out by Parmee and Hajela [2012], *numerical optimization is now a mature technology,* that is, fast, robust, and capable of solving problems with up to millions of variables, that can be used in many aspects of astronomy. The development of an optimization application should be performed according to the following steps. First formalize the problem by identifying the variables, the objective function and the constraints. Then restate and identify the nature of the problem (convex, differentiable, LP, QP, ...) to facilitate the use of state-of-the-art open-source optimization software tools such as CVX, SeDuMi, GLPK (for LP), openOpt (in python), Optaplanner (in java) or commercial ones such as Gurobi, Cplex, Mosek or Xpress to name a few[8]. Last, but not least, evaluate the solution provided and improve the model if necessary.

At this point it may be useful to provide the reader who wants more in-depth information with a list of specific references.

Convex optimization (with or without constraints) is a well investigated domain and we refer the reader to Boyd and Vandenberghe [2004] for a very pedagogical introduction, and to the books of Bertsekas [1999] and Nocedal and Wright [2006] for a different insight on convex optimization. The book Bubeck [2015], available on the author's website, also discusses conditional gradient and stochastic optimization often used in large scale problems.

Non-differentiable optimization using proximal algorithms has been treated extensively in Combettes and Pesquet [2011] and more recently in Parikh and Boyd [2014]. For a more detailed study of algorithms with sparsity inducing regularization we recommend Bach et al. [2011]. Finally, the convergence and theory of proximal (and monotone) operators is discussed in Bauschke and Combettes [2011].

No doubt that the future of optimization will also provide tools to efficiently handle new classes of problems not treated here, such as mixed integer programs.

References

A. Agnello, B. C. Kelly, T. Treu, and P. J. Marshall. Data mining for gravitationally lensed quasars. *Monthly Notices of the Royal Astronomical Society*, 448(2):1446–1462, 2015.

A. Antoniadis, I. Gijbels, and M. Nikolova. Penalized likelihood regression for generalized linear models with non-quadratic penalties. *Annals of the Institute of Statistical Mathematics*, 63(3):585–615, 2011.

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: an approach based

[8]For more details and comparisons see `plato.asu.edu/bench.html`.

on the kurdyka-lojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 5, 2011.

Bacon, R., Brinchmann, J., Richard, J., Contini, T., Drake, A., Franx, M., Tacchella, S., Vernet, J., Wisotzki, L., Blaizot, J., Bouché, N., Bouwens, R., Cantalupo, S., Carollo, C. M., Carton, D., Caruana, J., Clément, B., Dreizler, S., Epinat, B., Guiderdoni, B., Herenz, C., Husser, T.-O., Kamann, S., Kerutt, J., Kollatschny, W., Krajnovic, D., Lilly, S., Martinsson, T., Michel-Dansac, L., Patricio, V., Schaye, J., Shirazi, M., Soto, K., Soucail, G., Steinmetz, M., Urrutia, T., Weilbacher, P., and de Zeeuw, T. The muse 3d view of the hubble deep field south. *A&A*, 575:A75, 2015. doi: 10.1051/0004-6361/201425419. URL `http://dx.doi.org/10.1051/0004-6361/201425419`.

N. M. Ball and R. J. Brunner. Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(07):1049–1106, 2010.

J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media, 2011.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

D. P. Bertsekas. Nonlinear programming. 1999.

L. Bottou. Stochastic learning. In *Advanced lectures on machine learning*, pages 146–168. Springer, 2004.

S. Bourguignon, D. Mary, and E. Slezak. Sparsity-based denoising of hyperspectral astrophysical data with colored noise: Application to the muse instrument. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, pages 1–4. IEEE, 2010.

S. Bourguignon, D. Mary, and É. Slezak. Processing muse hyperspectral data: Denoising, deconvolution and detection of astrophysical sources. *Statistical Methodology*, 9(1):32–43, 2012.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The annals of applied statistics*, 5(1):232, 2011.

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

R. Carrillo, J. McEwen, and Y. Wiaux. Sparsity averaging reweighted analysis (sara): a novel algorithm for radio-interferometric imaging. *Monthly Notices of the Royal Astronomical Society*, 426(2):1223–1234, 2012.

R. E. Carrillo, J. D. McEwen, and Y. Wiaux. Purify: a new approach to radio-interferometric imaging. *Monthly Notices of the Royal Astronomical Society*, 439(4):3591–3604, 2014.

J. Christensen-Dalsgaard. *Lecture notes on stellar oscillations,* `http://astro.phys.au.dk/~jcd/oscilnotes/`, 2003.

J. Christensen-Dalsgaard. *Lecture notes on stellar evolution,* `http://astro.phys.au.dk/~jcd/evolnotes/LN_stellar_structure.pdf`, 2008.

F. H. Clarke. *Optimization and nonsmooth analysis*, volume 5. Siam, 1990.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

T. J. Cornwell. Multiscale clean deconvolution of radio synthesis images. *Selected Topics in Signal Processing, IEEE Journal of*, 2(5):793–801, 2008.

A. R. De Pierro. On the convergence of the iterative image space reconstruction algorithm for volume ect. *IEEE TRANS. MED. IMAG.*, 6(2):174–175, 1987.

P. Gong, C. Zhang, Z. Lu, J. Z. Huang, and J. Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Machine learning: proceedings of the International Conference. International Conference on Machine Learning*, volume 28, page 37. NIH Public Access, 2013.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

M. Grant, S. Boyd, and Y. Ye. *Disciplined convex programming*. Springer, 2006.

T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

J. Högbom. Aperture synthesis with a non-regular distribution of interferometer baselines. *Astronomy and Astrophysics Supplement Series*, 15:417, 1974.

Ž. Ivezić, A. K. Vivas, R. H. Lupton, and R. Zinn. The selection of rr lyrae stars using single-epoch data. *The Astronomical Journal*, 129(2):1096, 2005.

Ž. Ivezić, A. J. Connolly, J. T. VanderPlas, and A. Gray. *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data: A Practical Python Guide for the Analysis of Survey Data*. Princeton University Press, 2014.

N. Komodakis and J.-C. Pesquet. Playing with duality: An overview of recent primal? dual approaches for solving large-scale optimization problems. *Signal Processing Magazine, IEEE*, 32(6):31–54, 2015.

H. Lantéri, C. Theys, and C. Richard. Constrained minimization algorithms. *EAS Publications Series*, 59:303–324, 2013.

D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

L. B. Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974.

T. Mitchell. *The Discipline of Machine Learning*, 2006.

Y. Nesterov et al. Gradient methods for minimizing composite objective function. Technical report, UCL, 2007.

J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

I. Parmee and P. Hajela. *Optimization in industry*. Springer Science & Business Media, 2012.

M. Perryman. *The exoplanet handbook*. Cambridge University Press, 2011.

W. H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.

A. J. Smola and B. Schölkopf. *Learning with kernels*. Citeseer, 1998.

J.-L. Starck, F. Murtagh, and J. Fadili. *Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis*. Cambridge University Press, 2015.

R. F. R. Suleiman, D. Mary, and A. Ferrari. Dimension reduction for hypothesis testing in worst-case scenarios. *IEEE Trans. Signal Processing*, 62(22):5973–5986, 2014. doi: 10.1109/TSP.2014.2359641. URL `http://dx.doi.org/10.1109/TSP.2014.2359641`.

M. Süveges et al. A comparative study of four significance measures for periodicity detection in astronomical surveys. *MNRAS*, 450(2):2052–2066, 2015.

D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

A. R. Thompson, J. M. Moran, and J. George W Swenson. *Interferometry and Synthesis in Radio Astronomy*. John Wiley & Sons, Nov. 2008.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, Series B (Methodological)(58(1)):267–288, 1996.

B. Wakker and U. Schwarz. The multi-resolution clean and its application to the short-spacing problem in interferometry. *Astronomy and Astrophysics*, 200:312–322, 1988.

S. J. Wright, R. D. Nowak, and M. A. Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7): 2479–2493, 2009.

Z. Xu, X. Chang, F. Xu, and H. Zhang. $L_{1/2}$ regularization: a thresholding representation theory and a fast solver. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7):1013–1027, 2012.

D. G. York, J. Adelman, J. E. Anderson Jr, S. F. Anderson, J. Annis, N. A. Bahcall, J. Bakken, R. Barkhouser, S. Bastian, E. Berman, et al. The sloan digital sky survey: Technical summary. *The Astronomical Journal*, 120(3): 1579, 2000.